

Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Autentizace v systému eLogika pomocí NFC
NFC Authentisation in eLogika System

2014

Daniel Šimera

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Daniel Šimera**
Studijní program: B2647 Informační a komunikační technologie
Studijní obor: 2612R025 Informatika a výpočetní technika
Téma: Autentizace v systému eLogika pomocí NFC
NFC Authentisation in eLogika System

Zásady pro vypracování:

Cílem práce je navrhnout a naimplementovat systém, který za pomoci NFC provede identifikaci studentů v systému eLogika. Systém bude naimplementován pro platformu Windows phone 8.

Práce bude obsahovat:

1. Rozebrání problematiky NFC, možnosti využití pro identifikaci, rozebrání právních aspektů využití a ukládání s právní normou.
2. Analýzu, návrh a implementaci systému, který provede identifikaci v systému eLogika.
3. Analýzu využití NFC pro logování do systému eLogika.
4. Návrhy na další možné využití NFC v systému eLogika.

Seznam doporučené odborné literatury:

Vedat Coskun, Kerem Ok, Busra Ozdenizci: Near Field Communication (NFC): From Theory to Practice, ISBN 978-1-119-97109-2

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Mgr. Marek Menšík, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka
vedoucí katedry

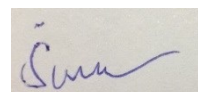


prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 5. května 2014



.....
podpis studenta

Poděkování

Rád bych poděkoval Mgr. Marku Menšíkovi, Ph.D. za odbornou pomoc a konzultaci při vytváření této bakalářské práce.

Abstrakt

Cílem této bakalářské práce je rozšířit možnosti studenta při autentizaci na testech a aktivitách systému eLogika. K autentizaci bude využito aplikace určené pro operační systém Windows Phone 8. Práce v úvodu popisuje systému eLogika a jeho funkcionalitu. Dále jsou rozebrány bezdrátové komunikační technologie NFC (Near Field Communication) a RFID (Radio Frequency Identification, které bude aplikace využívat. Popsán je i samotný mobilní operační systém Windows Phone. Následně je proveden funkční návrh, analýza a implementace aplikace, která je v poslední části práce představena.

Klíčová slova

Near Field Communication, eLogika, Autentizace, Windows Phone 8, Extensible Application Markup Language, C#, Windows Communication Foundation

Abstract

The goal of this thesis is to extend students possibilities while authenticating before exams and other activities managed by eLogika system. An application for operating system Windows Phone 8 is used for authentication. In the introduction, system eLogika and its features are described. In the next part, wireless communication protocols such as NFC (Near Field Communication) and RFID (Radio Frequency Identification) are analyzed. In the end of the theoretical part Windows Phone operation system is described. In the practical part, the application design and its implementation specifics are presented.

Key words

Near Field Communication, eLogika, Authentication, Windows Phone 8, Extensible Application Markup Language, C#, Windows Communication Foundation

Seznam použitých zkratk a symbolů

ARM	Advanced RISC Machine
CRL	Center for Research Libraries
GPU	Graphic Processing Unit
HTTP	Hypertext Transfer Protocol
LMS	Learning Managment Systém
MVC	Model View Controller
MVVM	Model View ViewModel
NFC	Near Field Communication
RFID	Radio Frequency Identification
SDK	Software Development Kit
TCP/IP	Transmisson Control Protocol/Internet Protocol
VGA	Video Graphics Array
WCF	Windows Communication Foundation
WVGA	Wide Video Graphics Array
XAML	Extensible Application Markup Language

Obsah

1	Úvod.....	- 1 -
1.1	Cíl práce	- 1 -
2	Systém eLogika.....	- 2 -
2.1	Uživatelské role	- 2 -
2.2	Můj přínos a využití systému eLogika	- 4 -
3	Použité technologie	- 6 -
3.1	NFC.....	- 6 -
3.2	RFID.....	- 9 -
4	Operační systém Windows Phone.....	- 11 -
4.1	Vývoj aplikací.....	- 13 -
4.2	Publikace aplikace na Windows Store	- 15 -
5	Specifikace a analýza aplikace.....	- 16 -
5.1	Specifikace aplikace.....	- 16 -
5.2	Funkce aplikace	- 16 -
5.2.1	Funkce role student.....	- 16 -
5.2.2	Funkce role tutor/garant.....	- 17 -
5.3	Analýza.....	- 18 -
5.4	Komunikace se systémem eLogika	- 19 -
5.5	Komunikace s Raspberry PI	- 21 -
5.6	Zvolené technologie	- 22 -
6	Aplikace a její implementace.....	- 23 -
6.1	Přihlašování	- 23 -
6.2	Rozhraní role student	- 25 -
6.2.1	Rozhraní fotoaparátu	- 26 -
6.3	Rozhraní role tutor/garant.....	- 28 -
	Závěr	- 31 -
	Seznam obrázků.....	- 32 -
	Seznam tabulek.....	- 33 -
	Seznam příloh.....	- 34 -
	Použitá literatura.....	- 35 -

1 Úvod

V současné době se informační technologie stávají čím dál tím víc neodmyslitelnou součástí života lidí. Ať už je to vědomě nebo nevědomě, každý den je používáme a ulehčujeme si pomocí nich život.

Počítače se vyskytují v různých podobách. V minulosti zabíraly celé místnosti a nyní, v případě mobilních telefonů, se nám vměstňají do dlaně. S možností vytvořit počítač o takové velikosti a konfiguraci, aby bylo možné ji nosit stále s sebou, vnikla nová oblast informačních technologií zvaná mobilní technologie, která nám otevřela zcela nové možnosti využití počítačů a propojila ji s mnoha jinými technologiemi.

Momentální situace nám ukazuje, že počítače malé velikosti mají největší zastoupení zejména v mobilních telefonech. Zároveň je nutno podotknout, že se nejedná jen o mobilní telefon, ale jsou s ním spojeny další užitečné funkce jiných zařízení, jako fotoaparát, multimediální přehrávač, navigace a další. Tato zařízení poskytují výpočetní výkon, o kterém se nám v minulosti ani nezdálo a to samé platí i o jejich paměti. Tyto aspekty jim zaručí široké spektrum využití. Jedním příkladem využití je bezdrátová technologie NFC.

Integrací této technologie do mobilních telefonů se jejich možnosti ještě více rozšiřují. Vůbec první myšlenkou NFC bylo integrování osobních objektů do elektronických zařízení a tak eliminovat potřebu nosit čím dál tím větší množství autorizačních a autentizačních objektů fyzicky s sebou. Po umístění NFC zařízení na fyzický objekt o něm dokážeme jednoduchým dotykem získat dodatečné informace potřebné k vykonání libovolné akce a tímto propojit fyzický svět s elektronickým. Ideálním vzorovým příkladem může být připojení k bezdrátové síti Wi-Fi, kdy se mobilním telefonem a dotkneme NFC čipu, který v sobě obsahuje konfiguraci daného přístupového bodu, a připojení je tak velmi jednoduše zprostředkováno.

1.1 Cíl práce

Cílem mé bakalářské práce je vyvinout mobilní aplikaci pro mobilní operační systém Windows Phone 8, která bude mít za úkol provádět pomocí NFC autentizaci v systému eLogika. Autentizace se bude provádět na aktivitách (testy, projekty apod.) systému eLogika, kde jak student, tak vyučující bude mít možnost k autentizaci využít právě mé aplikace.

V první části se věnuji samotné technologii NFC a jejímu využití pro identifikaci. V neposlední řadě také rozebrání právních aspektů využití a ukládání s právní normou. S ohledem na tyto faktory provedu analýzu, návrh a implementaci samotné aplikace.

2 Systém eLogika

Má aplikace ke svému běhu využívá systému eLogika, což je řídicí výukový systém, který je postaven na originální terminologii Learning Management System (LMS).

Mezi základní funkce systému LMS patří vytváření a následná kompletní správa kurzů, evidence studentů, výukových materiálů, apod.

Systém eLogika se prezentuje jako webová aplikace využívající e-learningovou formu vzdělávání, která nám při použití moderních komunikačních technologií přináší řadu výhod. Hlavní výhodou je nezávislost studenta na čase a místu dané výuky, dále jednoduchý přístup ke studijním materiálům a průběžné testování znalostí. Tyto výhody nám však vytváří i nevýhody. Jednou z nich je omezení osobního kontaktu mezi studentem a vyučujícím, která s sebou přináší delší prodlevu při řešení problémů s probíranou látkou. Další nevýhodou může být také nutná dostupnost internetu a zařízení potřebné k přístupu do e-learningu.

Technologie

eLogika využívá .NET Framework 4.5, konkrétně technologii ASP.NET což je produkt společnosti Microsoft Corporation. Hlavní výhodou a taky důvodem zvolení této technologie je prostředí CRL, které je společné i pro další technologie .NET Frameworku.

Pro úschovu dat se využívá relační databázový systém Microsoft SQL Server 2012 ve verzi Enterprise, což je nejvyšší možná varianta disponující nadstandardními funkcemi, které se u nižších variant nevyskytují. Jde např. o funkce data miningu a zabezpečení.

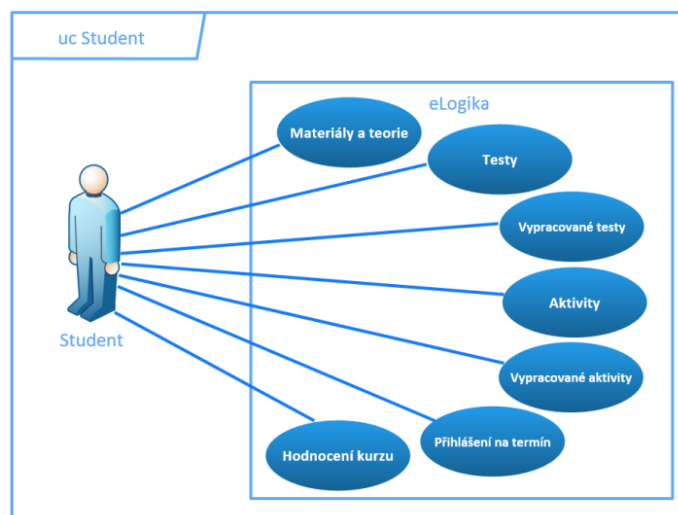
Server využívá operační systém Microsoft Windows Server 2008R2, který byl zvolen s ohledem na výše zmíněné technologie. Součástí operačního systému je také webový server ISS 7.0, který se stará běh všech webových aplikací.

2.1 Uživatelské role

Uživatel systému eLogika se stává každý, kdo má v systému vytvořen svůj vlastní účet. Aby se však uživatel mohl dostat k jednotlivým funkcím a možnostem systému, musí vlastnit alespoň jednu z celkových pěti uživatelských rolí, které systém eLogika definuje. Každý uživatel může vlastnit jednu a více takových rolí. Současně však může pracovat jen s jednou ze svých rolí, kterou si zvolí při přihlašování do systému. Moje mobilní aplikace bude pracovat se třemi z těchto rolí, které bych rád v dalších bodech popsal. Jedná se o role student, garant a role tutor.[1]

Role student

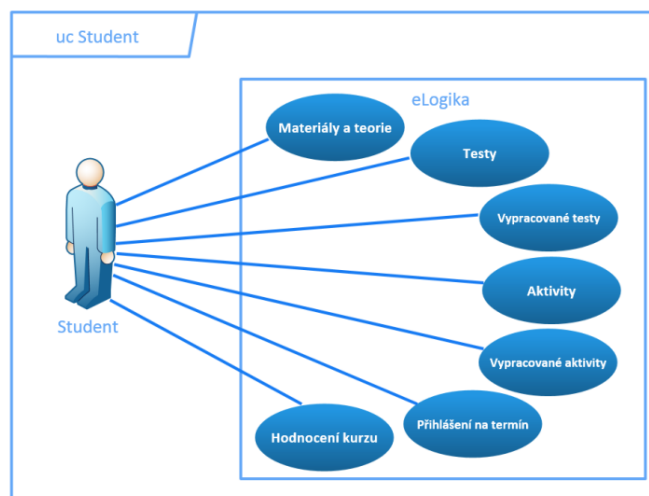
I když je student v rámci uživatelských práv v systému ten nejméně zodpovědný, tak se dá označit za hlavní roli systému, jelikož právě studenti jsou ti, kteří budou do systému nejčastěji přistupovat. Jeho funkce a možnosti odpovídají běžným studentským činnostem při klasické výuce. Zahrnují např. vykonávání testu či dalších různých aktivit, přihlašování na testy, náhledy do studijních materiálů, apod.



Obrázek 2.1: Funkce role student

Role tutor

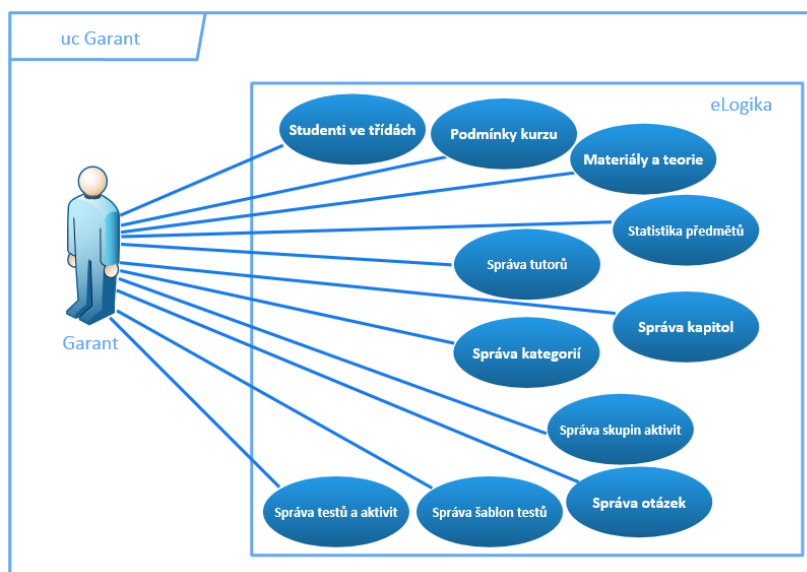
Uživatel, který disponuje touto rolí, se dá přirovnat k vyučujícímu (cvičícímu) konkrétní skupiny studentů. Tato skupina studentů je v rámci eLogiky pojmenována jako třída. Funkce tutora spočívají v přidělování studijních aktivit dané skupině, přípravě testů a dále třeba hodnocení výsledků.



Obrázek 2.2: Funkce role tutor

Role garant

Garant zodpovídá za celý předmět (kurz), který studenti můžou absolvovat. Vytváří a definuje jeho náplň a podmínky k jeho vykonání. Dále se garant podílí na tvorbě zkouškových testů, které slouží k závěrečnému prověření studenta a jeho nabytých vědomostí.



Obrázek 2.3: Funkce role garant

Ostatní role

Systém dále definuje ještě role tajemník a administrátor, které sice v mé aplikaci neúřadují, ale bylo by dobré si je také popsat.

Uživatel v roli tajemníka má největší pravomoci v rámci dané školy (univerzity) a jeho povinností je připravit prostředí pro výuku v konkrétním akademickém roce. Musí vytvořit strukturu daného roku, tedy připravit rozdělení na potřebný počet časových úseků, pro které jsou pak definovány kurzy, které se v daném úseku budou vyučovat. Po vytvoření takové struktury je pak tajemník zodpovědný za přiřazení garantů k jednotlivým kurzům.

Role administrátor vlastní v rámci celého systému eLogika to nejvyšší oprávnění. V jeho kompetenci je správa škol, které mají udělenou potřebnou licenci pro práci se systémem. Dále role administrátora jako jediná povoluje kompletní správu jednotlivých účtů v systému.

2.2 Můj přínos a využití systému eLogika

Hlavním přínosem mé práce je rozšířit možnosti uživatelů systému při samotném ověřování identity na studijních aktivitách. Momentálně je student odkázán pouze na svoji studijní kartu a to by se mělo s příchodem mé aplikace zlepšit. Přispívá k tomu fakt, že mobilní telefon má u sebe snad každý, kdežto u studijních karet to bývá daleko horší.

Aplikace však ulehčí práci jen uživatelům určité uživatelské role. Bude k dispozici pro studenta, tutora a garanta. Pro další role z důvodů využitelnosti nemá žádný význam.

Co se týče informací, ze kterých aplikace čerpá, tak to jsou zejména informace o jednotlivých uživateli (přihlašovací jméno). Dále u studenta fotografie a u tutora/garanta termíny jednotlivých aktivit, ze kterých si má možnost vybrat.

Rozebrání právních aspektů

Z hlediska využití mé aplikace je zřejmé, že se při jejím běhu zpracovávají údaje studenta, konkrétně jeho fotografie a přihlašovací jméno, proto považuji za nutné vyhodnotit tyto údaje také z pohledu ochrany práv studenta. Přičemž vycházím ze zákona o ochraně osobních údajů č. 101/2000 Sb (dále jen OOU) a nového občanského zákoníku (dále jen NOZ).

Vycházím-li ze zákona o ochraně osobních údajů dle § 4 písmeno a, kde se definuje osobní údaj jako takový, ze kterého lze na základě přihlašovacího jména a fotografie identifikovat konkrétní osobu, je nutno respektovat povinnosti zpracovatele (má aplikace a systém eLogika) a seznámit uživatele (studenta) s důvodem uchování dat a zohlednit to, že osobní údaje mohou být zpracovány pouze se souhlasem uživatele, jak uvádí § 5 odstavec 2 OOU a § 84 NOZ.

Dle tvrzení stejných zákonů jsem sice zjistil, že svolení uživatele není třeba, jestliže zprostředkovatel (aplikace) provádí zpracování nezbytné nutné pro dodržení svých povinností, jak je uvedeno v § 5/2 a § 88 NOZ: "Subjekt údajů musí být při udělení souhlasu informován o tom, pro jaký účel zpracování a k jakým osobním údajům je souhlas dáván, jakému správci a na jaké období. Souhlas subjektu údajů se zpracováním osobních údajů musí být správce schopen prokázat" po celou dobu zpracování."

Považuji však za nereálné ukládat fotografii, proti vůli uživatele. Pokud student souhlas k uložení fotografie neudělí, postrádá využití mé aplikace smysl. Proto budu vycházet z toho, že student bude souhlasit a musí být poučen o tom, pro jaký účel je souhlas udílen, tedy autentizace studenta na testech a aktivitách systému eLogika.

Vzhledem k tomu, že fotografie a přihlašovací jméno je možno v mé aplikaci použít jen při samotné autentizaci, nepovažuji za nutné přijímat opatření ve smyslu § 13 OOU, který stanovuje povinnosti osob při zabezpečení osobních údajů. To samé platí i pro ochranu práv subjektu údajů.

3 Použité technologie

3.1 NFC

Bezdrátová technologie fungující na bázi rádiového přenosu dat na krátkou vzdálenost, která umožňuje jednoduchou a bezpečnou obousměrnou komunikaci mezi elektronickými zařízeními. Technologie NFC kombinuje funkci inteligentního klíče a NFC čtečky, a tak nám umožní snadný přístup k různým službám. Mezi nejaktuálnější využití NFC patří především možnost provádět bezkontaktní transakce. Dále to může být třeba přístup k digitálnímu obsahu a to vše probíhá za pomoci jediného jednoduchého dotyku.

- technologie NFC má pro sebe vyhrazené pásmo a pracuje na frekvenci 13,56 MHz
- rychlost přenosu dat je 106, 212 nebo 424 kbit/s
- data jsou přenášena tzv. induktivní vf vazbou (jedná se o rádiový přenos)
- přenos je half-duplex
- podporuje dva módy a to aktivní a pasivní
- používá vlastní Open Source protokol NFC

Při komunikaci se můžeme setkat s dvěma způsoby přenosu:

Aktivní přenos

Znamená, že obě strany spolu komunikují a navzájem se střídají ve směru přenosu signálu.

Pasivní přenos

Tento způsob spočívá v tom, že jedna strana je napájena elektromagnetickým polem vysílače, tedy nepotřebuje ke své práci žádné napájení. Zařízení je pak velmi levné.

Historie

NFC (Near Field Communication) není žádnou žhavou novinkou. O tomto standardu se mluví již několik let, teprve nyní se ale konečně dostává do prvních reálně dostupných přístrojů. Spolu s nimi se také pomalu, ale jistě, objevují způsoby, jak jej využít. A to nejen ve světě, ale i v ČR.

O samotné technologii NFC se více začíná hovořit v roce 2004, kdy společnosti Nokia, Philips a Sony ustavili tzv. Near Field Communication Forum, které dnes stojí za rozvojem a šířením této technologie. Prvním přístrojem s podporou NFC byla Nokia 6131 z roku 2006.[2]

Standardy

NFC má pod svými křídly hned několik komunikačních protokolů a formátů popisujících přenášená data. Je založena na standardech RFID zahrnující ISO/IEC 14443 a FeliCa. Tyto standardy jsou součástí ISO/IEC 18092 definovaném neziskovou organizací NFC Forum, jež byla založena v roce 2004 firmami NOKIA, Philips a Sony.

Aplikace bude komunikovat mezi dvěma mobilními zařízeními. V této variantě se klasicky používá tzv. Peer-to-peer režim, kdy je využito aktivního datového přenosu.

Rozhraní komunikačního peer-to-peer režimu, který umožňuje obousměrnou komunikaci mezi NFC zařízeními, je definováno v rámci standardu NFCIP-1. Tento režim je určen především pro vzájemnou výměnu dat, v našem případě textového řetězce s identifikací daného studenta. U tohoto režimu se předpokládá, že veškerá zařízení budou v aktivním režimu během komunikace. Maximální přenosové rychlosti dosahují 424 kbit/s.

Standard NFCIP-1 poskytuje na základní funkce linkové vrstvy, jako opravy chyb, potvrzování rámců, jejich řazení a další vlastnosti. LLCP protokol rozšiřuje základní funkcionalitu standardu NFCIP-1 o další důležité služby - spojitě orientovaný transport rámců, nespojitě orientovaný transport (nepotvrzovaný), řízení stavu linky, asynchronní vyvažovanou komunikaci a multiplexaci protokolů.[3]

Možná využití NFC

- nahrazení debetních či čipových karet)
- prodej jízdenek na veřejnou dopravu
- Bluetooth a Wi-Fi připojení
- párování zařízení přes NFC a následná komunikace a přenos dat proběhne prostřednictvím Bluetooth
- NFC může být také využito k tomu, aby nám nastavilo Wi-Fi připojení dle daného přístupového bodu, což může být pro řadu uživatelů přínosem, zvláště u připojování k pokročileji nastaveným či zabezpečeným sítím (zejména na univerzitách)
- Identifikace
- náhrada stávajícího RFID, které oproti NFC nemá tolik zabezpečení

NFC tagy

Zástupce pasivních NFC zařízení, který se může jevit jako mála bezdrátová paměťová karta. Na rozdíl o nich má "tag" výrazně menší kapacitu. I ti největší zástupci mají kapacitu v jednotkách kilobajtů.

Využití NFC tagů

- Jízdenky a předplacené kupony pro MHD
- Občanské a řidičské průkazy, pasy
- Docházkové a věrnostní karty
- Klíče, vstupní karty a žetony
- Vizitky
- Reklama

Srovnání s příbuznými technologiemi

Nabízí se i logická otázka porovnání s ostatními bezdrátovými technologiemi, které však nemá NFC nahradit, ale má vedle nich spíše stát jako rovnocenný partner s přece jen jinými možnostmi a úkoly.

Technologie známé zejména z počítačového světa (Bluetooth, Wi-Fi, Wireless USB apod.) vznikly za jiným účelem – totiž za účelem přenosu větších objemů dat, případně kvůli možnosti bezdrátového připojení periférií. Na rozdíl od NFC neumožňují tak vysoké přizpůsobení koncových aplikací, provázání se SIM kartou a celkově zamýšlenému účely zdaleka nevyhovují tak dobře.

Technologie typu RFID a bezkontaktních (smart) karet jsou si již s NFC mnohem bližší a mnohé mají společné. Všechny tyto technologie mají podobné využití a fungují podobně, NFC je ale novější standard, nabízí více možností a zejména napojení na mobilní telefony (respektive SIM karty).[4]

	<i>NFC</i>	<i>RFID</i>	<i>Irda</i>	<i>Bluetooth</i>
<i>Doba nastavení</i>	<i><0,1ms</i>	<i><0,1ms</i>	<i>~0,5s</i>	<i>~6sec</i>
<i>Dosah</i>	<i>do 10cm</i>	<i>do 3m</i>	<i>do 5m</i>	<i>do 30m</i>
<i>Zabezpečení</i>	<i>Vysoké</i>	<i>Částečné</i>	<i>-</i>	<i>Kdo jsi?</i>
<i>Použití</i>	<i>Platby, získání přístupu, sdílení</i>	<i>Identifikace při sportovních akcích</i>	<i>Dálkové ovládání</i>	<i>Výměna dat</i>

Tabulka 1: Srovnání bezdrátových komunikačních technologií

Výhody

- Jednoduchost – navázání spojení je otázkou přiblížení obou přístrojů
- Všestrannost – NFC je možné využít v nejrůznějších prostředích a při nejrůznějších situacích
- Otevřenost a standardy – základní NFC vrstvy vycházejí z otevřených standardů
- Propojitelnost – NFC je možné využít např. i pro komunikaci Wi-Fi a Bluetooth zařízení
- Bezpečnost – protože přenosy probíhají na velmi krátké vzdálenosti, je velmi těžké je zachytit a zneužít, NFC navíc dovoluje využívat zabezpečené aplikace
- Kompatibilita – NFC je možné využít s některými stávajícími technologiemi bezkontaktních karet

Nevýhody

- Rozšíření - v současné době je NFC přidáváno jen do lepších mobilních telefonů
- Obavy uživatelů - ztrátu telefonu můžeme přirovnat k ztrátě peněženky, tudíž to řadě uživatelů přináší obavy

3.2 RFID

Technologii RFID (Radio Frequency Identification) můžeme označit za předchůdce NFC. Jedná se o identifikaci pracující ve vysokofrekvenčním pásmu.

Ačkoliv si to možná nepřipouštíme, setkáváme se s touto technologií takřka denně. Technologie RFID je součástí našeho života, která zvyšuje produktivitu a pohodlí.

Příklady využití

- Prevence proti krádežím v obchodech
- Nákup jízdenek v MHD
- Docházkový systém firem
- Výběr dálničních poplatků

Historie

Počátky sahají až do období druhé světové války. V roce 1939 britská armáda využila systém zvaný IFF (Identify: Friend or Foesystem). Sloužil k rozpoznávání vlastních letadel od nepřátelských. Na palubě britského letounu byl umístěn vysílač, který po přijetí signálu ze země odvysílal signál, na základě kterého mohl operátor na zemi rozpoznat, o jaký letoun se jedná. Systém fungoval na principu RFID.

Jako pravý prvopočátek se ale považuje datum 23. ledna 1970, kdy Mario W. Cardullo obdržel první patent pro aktivní RFID tagy. Ještě v téže roce získal kalifornský podnikatel Charles Walton patent pro pasivní tagy, které používal pro odemykání dveří. Okolo roku 1977 se technologie poprvé dostala do veřejného sektoru. Na začátku se využívala hlavně při sledování dobytka. Další vývoj RFID technologie velice pozvolný a probíhal zejména v laboratorních podmínkách. Vývojem se zabývali firmy jako IBM, Fairchild a ComServ.

Největší tlak pro zavedení RFID do běžného života přišel od společnosti WalMart, jenž se zabývala používáním čárových kódů. Cílem společnosti bylo vyvinout technologii, která v reálném čase dokáže identifikovat větší počet objektů na delší vzdálenost a to bez nutnosti přímé viditelnosti, což při využití čárových kódů nebylo možné. Čárový kód totiž vyžaduje přímou viditelnost s čtecím zařízením.[5]

Architektura

Technologie RFID využívá při komunikaci, nosiče dat (RFID tag) a čtecím zařízením, elektromagnetické pole.[6]

Každý RFID systém se skládá ze čtyř částí:

- RFID tag
- Čtecí zařízení
- Middleware
- Databázové uložení dat

RFID tag

Co se týče architektury systémů, tak se tag nachází na nejnižší úrovni. Každý tag se skládá s mikročipu (elektronický paměťový obvod), který nese data a antény. Spolu jsou ukryty v kompaktním obale. V případě aktivního tagu je obsažen vlastní zdroj napájení.

Anténa tagu zachytí signál od čtecího zařízení a vrátí jej obvykle s dalšími údaji jako je například identifikační číslo nebo další upravené informace.

Čtecí zařízení

Slouží k získání informace z RFID tagu. Toto zařízení obsahuje anténu, která vyzařuje rádiové vlny do okolí a tag reaguje tak, že po získání energie z těchto elektromagnetických vln, zasílá své data zpět. Složitější čtečka může obsahovat další funkce jako například:

- Zajištění anti-kolizních opatření (R/W komunikace s více tagy současně)
- Šifrování dat (ochrana identity)
- Ověření tagů (zabránění neoprávněnému přístupu)

Na největší vzdálenost, při které se dá tag přečíst, může působit množství faktorů. Například frekvenční pásmo, výkon antén nebo orientace a polarizace antény.

RFID čtečka je obvykle napájena ze sítě osobního počítače, který ji také využívá. Pro připojení se nejčastěji využívá rozhraní RS232, USB nebo Ethernet. Velké množství výrobců používá pro komunikaci různé podporné knihovny a programové vybavení.

Middleware

Tato část systému RFID slouží jako prostředník informačním systémem a čtecím zařízením.

Výhody

- Bezkontaktní povaha technologie, nevyžadující pro svoji činnost ani přesné polohování jako je infračervený přenos dat
- Lze zaznamenávat, uchovávat a poskytovat informace v reálném čase
- Velká odolnost RFID čipů (vlhkost, teplota, atd.)
- Přenosu dat z čipu nebrání špatné optické či atmosférické podmínky
- Přínosem je i rychlost čtení, pohybující se většinou v časech pod 100 milisekund
- Aktivní čipy pak přináší nové možnosti interakce do identifikačního procesu
- Množství informací, které lze do čipu zapsat (cca 1MB)

Nevýhody

- Možnost sledování pomocí dostatečně výkonné čtečky
- Možnost naklonování RFID tagu

4 Operační systém Windows Phone

Hardwarové požadavky systému

Windows Phone je operační systém celosvětově známé společnosti Microsoft. Tento systém je určen pro chytré telefony o úhlopříčkách displeje od 3,8“ do 6“. Pro běh systému Microsoft jednoznačně stanovil minimální hardwarové požadavky, které musí mobilní telefon splňovat.[7]

- Displej (Kapacitní dotykový displej) – ovládání systému je založeno na gestech a dotycích. Některá gesta vyžadují tzv. multitouch (dotyk více prstů ve stejném čase) a jelikož starší rezistivní technologie toto neumožňuje, je kapacitní displej nutností. Takový displej však musí umožnit zpracovat minimálně čtyř dotyků, aby kritériím vyhověl.
- Procesor (2 jádrový procesor) – jelikož Windows Phone 8 nabízí plný multitasking, tak ke svému běhu potřebuje alespoň dvě výpočetní jádra procesoru, založeného na architektuře ARM.
- Operační paměť (512 MB RAM) – pro bezproblémový chod systému s WVGA (800x480) rozlišením je nutné mít aspoň 512 MB RAM a pro vyšší rozlišení je již vyžadováno 1024 MB operační paměti RAM.
- Uživatelská paměť (4 GB) – samotný systém si již po instalaci přivlastní cca 2 GB, pro aplikace a uživatelská data je tedy potřeba minimálně dalších 2 GB paměti.
- Fotoaparát – je nutno, aby byl přítomen fotoaparát minimálně s VGA (640x480) rozlišením na zadní straně přístroje. Přední fotoaparát není nutností.
- Hardwarová tlačítka – na každém zařízení musí nacházet minimálně 7 tlačítek. Zapínací tlačítko, spoušť fotoaparátu a tlačítka pro hlasitost musí být řešeny hardwarově. Tlačítko start, zpět a tlačítko pro hledání mohou být řešeny dotykovým provedením.
- GPU (podpora DirectX) – jelikož systém poskytuje nativní přístup k této knihovně, tak aplikace mohou naplno využívat hardwarové akcelerace Direct3D.

Již z výše uvedených minimálních požadavků vyplývá, že vývojář aplikací pro Windows Phone má jistotu, že uvedené senzory a další hardwarové vybavení mu budou vždy k dispozici.

Zástupci na trhu

Hlavním výrobcem mobilních telefonů s operačním systémem Windows Phone je bezesporu známa finská společnost NOKIA, která telefony Lumia pokrývá 90% trhu s telefony využívajícími tohoto systému. Dalšími výrobci jsou Samsung, HTC a Huawei, které představili jen velmi málo modelů svých chytrých telefonů se systémem Windows Phone a soustředí se na konkurenční platformy.

Historie verzí

Ačkoliv může číslování působit celkem nelogicky, první verze operačního systému Windows Phone nesla označení 7. Microsoft jí poprvé představil široké veřejnosti na Světovém mobilním kongresu (World Mobile Congress) v Barceloně 15. února 2010. Systém však byl oficiálně vydán až o necelých devět měsíců později, v listopadu 2010.

Systém byl vydán jako nástupce platformy Windows Mobile. Ke svému běhu využíval stejně jako jeho předchůdce jádro Windows CE. Jelikož však neexistovala žádná zpětná kompatibilita, tak jste s aplikací ze systému Windows Mobile na nové platformě neuspěli.

Vývoj této platformy dospěl k vydání řady aktualizací, z nichž ty největší nesly označení 7.5 (Mango) a později 7.8 (Tango). Kvůli rozdílnému jádru a potřebnému hardwaru nebylo možné žádné zařízení s Windows Phone 7 aktualizovat na verzi 8, což v uživatelské základně vzbudilo velkou vlnu nevole.

Tržní podíl Windows Phone 7 logicky klesá ve prospěch aktuálnější osmé verze. V současné době na trhu mobilních telefonů s Windows Phone zaujímá sedmá verze cca 25%.

Momentálně je postupně uvolňována verze označena jako Windows Phone 8.1, která poskytne uživatelům řadu novinek. Mezi ty nejzásadnější patří zakomponování notificačního centra do systému, dále možnost ovládat telefon hlasem (Cortana) a v poslední řadě také umístit za dlaždice vlastní tapetu.

Aktualizace na Windows Phone 8.1 je aktuálně dostupná pouze registrovaným vývojářům a majitelům nejnovějších zařízení. Postupně však bude zpřístupněna všem majitelům zařízení s Windows Phone 8.



Obrázek 4.1: Vývoj uživatelské rozhraní systému Windows Phone

Poskytované funkce

Operační systém Windows Phone poskytuje mimo běžných funkcí mobilního telefonu, jako je telefonování, psaní a správa SMS či správa kontaktů také další funkce, které se již dají považovat za standard.

Pro správu a synchronizaci e-mailů, kalendářů a kontaktů poslouží plná integrace s korporátním řešením Microsoft Exchange, kterou využijí nejen firemní zákazníci. Pro uživatele, kteří nepoužívají Exchange je připravena plná integrace služeb Outlook.com (dříve Hotmail) rovněž od Microsoftu, případně služeb Gmail společnosti Google.

Specifickou funkcí systému Windows Phone 8 je nativní podpora sociálních sítí Facebook, Twitter, MySpace a dalších. Integrace spočívá především v aktualizaci kontaktních informací, možnosti číst statusy jednotlivých kontaktů či sdílet vlastní, možnost prohlížet veškerá fotoalba svých kontaktů, apod.

Pokud některým uživatelům nevystačí interní paměť telefonu, mají v systému zajištěnu plnou podporu cloudového úložiště OneDrive. Funkce systému lze rozšířit pomocí aplikací třetích stran.

4.1 Vývoj aplikací

Architektura

Všeobecným doporučením při vývoji každé aplikace je oddělení jednotlivých vrstev aplikace a jejich vzájemná komunikace. V takovém případě se používá návrhových vzorů. Mezi nejvyužívanější patří MVC (Model View Controller). V případě vývoje pro platformu Windows Phone se jedná o MVVM (Model View ViewModel), který je však z MVC odvozen. Vývojáři však nejsou nuceni MVVM používat, jedná se pouze o doporučení, které vede k udržení vysoké kvality a hlavně přehlednosti zdrojového kódu.[7]

Model View ViewModel

MVVM patří mezi návrhové vzory založené na principu oddělení uživatelského rozhraní od aplikační logiky. Cílem takového vzoru je redukování velkého množství kódu a následné zjednodušení práce při jeho editaci.

- Model - reprezentuje data získané z persistentního úložiště dat (např. databáze). Model je dále zodpovědný za notifikace ViewModelu o změnách v datech.
- View - reprezentuje všechny prvky uživatelského rozhraní aplikace. Vrstva zodpovídá za zpracování uživatelského vstupu, které předává přímo ViewModelu
- ViewModel - zajišťuje přenos dat z Modelu k View a přenos uživatelských gest z View k Modelu.

Vývojové nástroje

K vývoji aplikací pro Windows Phone 8 se zpravidla používá vývojové prostředí Microsoft Visual Studio.

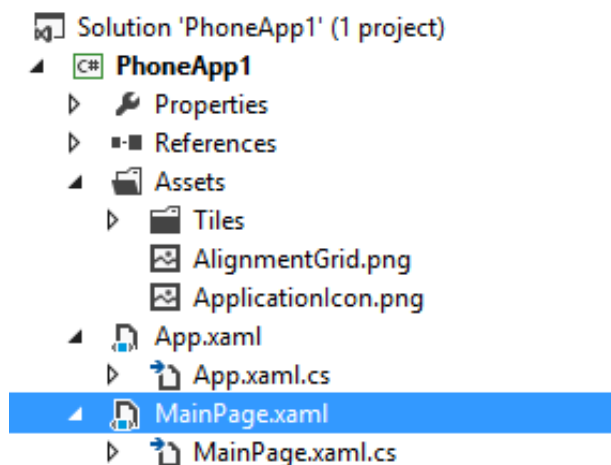
Základní instalace Visual Studia neposkytuje nástroje pro vývoj mobilních aplikací a je nutné je zvlášť doinstalovat. Jedná se o Windows Phone SDK.

Při tvorbě uživatelského rozhraní aplikace může vývojář využít samotného Visual Studia, jenž poskytuje základní nástroje a podporu jazyka XAML (více viz kapitola 5.8.1), ve kterém je uživatelské rozhraní vyvíjeno. V případě, že vývojář XAML neovládá, nabízí se možnost vytvoření uživatelské rozhraní v nástroji Expression Blend.[8]

Typy a struktura projektů

Po instalaci Windows Phone SDK se v průvodci přidání nového projektu ve Visual Studiu objeví nové skupiny projektů. Skupina Windows Phone nabízí následující projekty:

- Windows Phone App - prázdný projekt bez jakékoliv předpřipravené struktury
- Windows Phone Databound App - projekt, obsahující řádkové rozložení obsahu
- Windows Phone Panorama App - projekt, s parametrickým rozložením aplikace
- Windows Phone Panorama App - projekt, s tzv. Pivot rozložením aplikace
- Windows Phone HTML5 App - projekt, využívající internetového prohlížeče



Obrázek 4.2: Struktura projektu

- Properties - nachází se zde tzv. aplikační manifest, který obsahuje obecná nastavení aplikace. Jedná se o XML soubor, který lze jednoduše editovat
- Reference - zobrazí všechny referenční knihovny, jenž aplikace využívá
- Složka Assets - zde se umísťuje ikona aplikace a další grafické soubory aplikace
- App.xaml - obsahuje kód XAML hlavní stránky aplikace. Deklarují se zde prostředky, které se používají v aplikaci. Můžeme tento soubor požadovat za globální úroveň aplikace
- MainPage.xaml - kód XAML stránky aplikace, která se zorazí po jejím spuštění
- MainPage.xaml.cs, App.xaml.cs - kód na pozadí XAML stránky

Windows Store

Obchod s aplikacemi Windows Store je oficiální zdroj aplikací pro majitelé zařízení s Windows Phone a Windows 8. Na rozdíl od konkurenčních platforem, není zdroj těchto aplikací společný. Windows Store

Všechny aplikace na Windows Store museli být prověřeny a následně schváleny. Dále odpovídat požadavkům, které Microsoft stanovil. Právo k publikování aplikací mají pouze registrovaní vývojáři.

Tento postup praktikuje i konkurenční systém iOS od společnosti Apple. Naproti tomu nejmasovější mobilní operační systém Android společnosti Google nechává svůj obchod s aplikacemi dokořán všem vývojářům. Výhodou je větší počet aplikací, ale to vše jen na úkor kvality. Navíc se můžou objevovat i škodlivé aplikace, které můžou uživateli znepříjemnit život.[8]

4.2 Publikace aplikace na Windows Store

Jak už jsem zmínil výše, možnost publikovat své aplikace mají jen registrovaní vývojáři, kteří zaplatili roční aktivační poplatek ve výši \$99 USD. Studenti, kteří jsou zahrnuti do programu DreamSpark mají publikování aplikací zcela zdarma.

Schvalovací proces

Dříve než se propracuji k samotnému publikování aplikace, je třeba mít na paměti, že každá aplikace musí projít schvalovacím procesem, který se dělí do několika částí.

- Aplikace musí zákazníkům přinášet přidanou hodnotu
- Aplikace může zobrazovat reklamy, ale musí poskytovat více (funkcionality, přínosu), než jen zobrazení reklamy či zobrazení obsahu webové stránky
- Aplikace se musí chovat předvídatelným způsobem (musí poskytnout vizuální zpětnou vazbu, když se uživatel dotkne interaktivních prvků)
- Aplikace jsou řízeny uživatelem
- Aplikace musí být vhodné ke globálnímu nasazení
- Aplikace musí být snadno rozpoznatelné a pochopitelné

Pokud daná aplikace neprojde schvalovacím procesem, je vývojář vždy upozorněn, který z daných bodů nesplnil.

Zveřejnění aplikace

V případě úspěšného absolvování schvalovacího procesu má vývojář možnost určit, kdy bude aplikace zveřejněna. Pokud však této funkcionality využít nehodlá, má samozřejmě možnost zveřejnit aplikaci ihned.

Vývojář má také možnost využít zveřejnění aplikace v beta verzi. Taková aplikace však není veřejně přístupná. Bude přístupná jen těm uživatelům, které vývojář zvolí (až 1000 uživatelů). Taková varianta se hodí zejména při testování aplikace.[8]

5 Specifikace a analýza aplikace

5.1 Specifikace aplikace

- Využití jednotného přihlášení.
- Bude sloužit k autentizaci studentů na aktivitách systému eLogika
- Je vyvíjena na platformu mobilního operačního systému Windows Phone 8
- Využívá služeb systému eLogika
- Podmínkou využití aplikace je aktivní účet v systému eLogika
- Je vyžadováno internetové připojení, tudíž se bez něho uživatel neobejde
- Aplikace bude využívat čelní kamery a NFC čipu mobilního telefonu

Kdo bude s aplikací pracovat

V rámci aplikace se předpokládá víceuživatelský přístup. Uživatelé jsou nuceni mít zřízeny aktivní účet v systému eLogika, který bude disponovat přiřazením minimálně jedné uživatelské role.

Aplikace definuje tyto uživatelské role:

- Student
- Tutor
- Garant

5.2 Funkce aplikace

Po jednotném přihlášení se aplikace bude větvit dle uživatelské role, kde každému uživateli bude zobrazeno rozhraní, které si zvolil a má na něj příslušné právo. Funkce studenta jsou jednoznačné. Funkce tutora a garanta jsou identické, ale liší se v informacích, které jsou od systémem eLogika získány.

5.2.1 Funkce role student

Vytvoření fotografie:

Aby student mohl splnit podmínky autentizace, tak si musí vytvořit fotografii, která se připojí k jeho profilu. Může tak učinit následovně:

- Výběr z galerie mobilního telefonu
- Vytvoření fotografie v prostředí aplikace

Fotka je uložena jen do paměti aplikace. Chce-li jí student nahrát na sever, musí využít funkce schválení fotografie.

Autentizace:

Při prvním spuštění této funkce bude uživatel nucen absolvovat oba následující body. V dalších případech, pokud si to uživatel nevyžádá, se první bod nemusí provádět.

1. Schválení fotografie

Tato funkce očekává komunikaci s druhým zařízením pomocí čipu NFC. Druhá strana musí vyhovovat následujícím podmínkám:

- Zařízení musí vlastnit čip NFC
- Uživatel musí být přihlášen k roli tutor nebo garant

Pokud jsou podmínky splněny a komunikace proběhne korektně, bude studentovi umožněno nahrát fotografii na svůj profil.

2. Autentizace

Funkci bude možno spustit jen v momentě, kdy profil uživatele aplikace bude obsahovat schválenou fotografii. Je opět očekávána komunikace s druhým zařízením a je nutno počítat se stejnými podmínkami jako u výše zmíněné funkce. Při úspěšné komunikaci bude druhému zařízení předána informace o uživateli aplikace (přihlašovací jméno).

5.2.2 Funkce role tutor/garant

Výběr termínu:

Uživateli budou zobrazeny příslušné ovládací prvky, v kterých si musí postupně navolit příslušný termín, kde chce studenta autentizovat. Pořadí polí je následující:

- Výběr kurzu (např. Matematická logika)
- Výběr skupiny aktivity (např. zkouška, zápočet)
- Možnost výběru mezi testem a aktivitou (např. projekt)
- Výběr termínu testu nebo aktivity

Autentizace:

Využívá dat načtených z funkce výběrů termínu (identifikačním číslo zvoleného termínu). Uživatel si může vybrat z dvou variant (NFC, RFID).

1. NFC

Po zahájení a úspěšné komunikaci bude uživateli zobrazena fotografie aktuálně kontrolovaného studenta. Následně musí autentizaci potvrdit nebo zamítnout. Když potvrdí, bude dle zvoleného nastavení proveden tisk (QR kód, test).

2. RFID

Funkce spočívá v ověření studenta pomocí jeho studentské karty. Uživatel pouze stiskne tlačítko této funkce a následně bude informován o výsledku autentizace.

Schválení fotografie:

Počítá se s tím, že tuto funkci uživatel (vyučující) spustí až po překontrolování fotografie na straně studentské aplikace. Uživatel musí fotografii porovnat s fotografií na studentské kartě.

Dále očekává komunikaci s druhým zařízením pomocí čipu NFC. Druhé straně musí vyhovovat následujícím podmínkám:

- Zařízení musí vlastnit čip NFC
- Uživatel musí být přihlášen k roli student

Nefunkční požadavky

Pro svůj běh využívá aplikace webových služeb systému eLogika, ze kterých získává všechny potřebné informace. Korektní běh aplikace je tudíž závislý na serveru eLogiky.

Dále je zapotřebí přítomnost tiskového serveru, dostupného prostřednictvím mikropočítače Rabsberry PI.

5.3 Analýza

Funkcionalitu nejzásadnějších operací jsem se snažil nastínit v diagramu aktivit (uveden níže) a pomocí sekvenčních diagramů. (v příloze)

Proces přihlašování

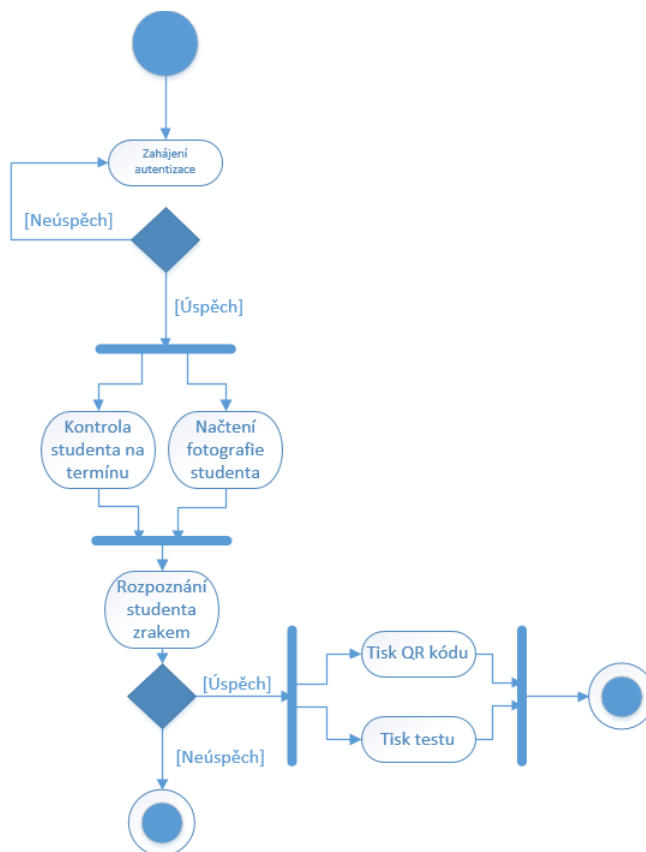
Tento proces se skládá z dvou bodů:

- Uživatel zadá přihlašovací údaje v podobě uživatelského jména a hesla, které jsou po stisku tlačítka překontrolovány serverem. Pokud je kontrola úspěšná, jsou uživateli zobrazeny univerzity, role a akademické roky, které má k dispozici.
- Uživateli si vybere požadované nastavení a poté je mu po stisknutí tlačítka spuštěno zvolené uživatelské rozhraní

Proces využívá webových služeb systému eLogika k ověření daného uživatele s databází uživatelů a na základě toho jsou pak do předpřipravené statické třídy uloženy všechny potřebné informace nutné pro další korektní běh aplikace.

Proces autentizace

Celý proces je započat v momentě, kdy mobilní zařízení vyučujícího obdrží od zařízení studenta přes NFC informaci o daném studentovi (přihlašovací jméno), které vyhledá na požadovaném termínu systému eLogika. Pokud se dopátrá k zdárnému konci, je vyučujícími zobrazena fotografie studenta, pomocí které musí studenta identifikovat. Pokud bude totožnost odpovídat, je po stisknutí tlačítka započat proces tisku, který zahrnuje jak tisk samotného testu, tak tisk QR kódu studenta. Viz tento aktivitní diagram nebo sekvenční diagram v příloze.



Obrázek 5.1: Aktivitní diagram autentizace

5.4 Komunikace se systémem eLogika

Veškerá komunikace mobilní aplikace se serverem eLogika probíhá pomocí WCF služeb, kterých je celá řada.

WCF

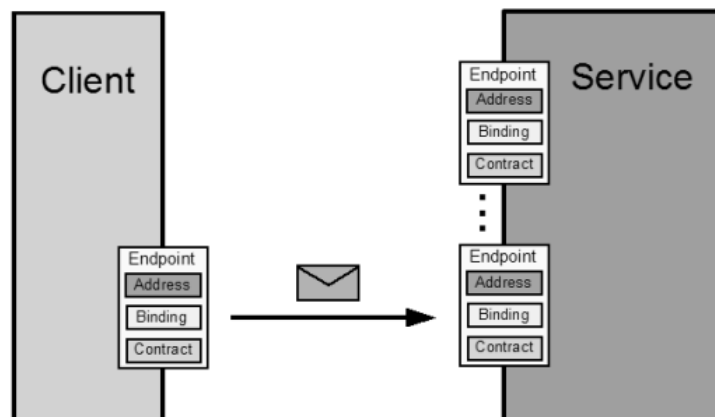
WCF (Windows Communication Foundation) služby jsou založeny na komunikaci pomocí zpráv.

Zpráva (message) je skupina dat obsahující záhlaví a tělo zprávy. Příkladem zprávy je v mém případě HTTP zpráva. Model WCF rozlišuje mezi klienty a službami. Klient je aplikace, která vyvolává komunikaci, kdežto služba čeká na požadavky od klientů.[9]

Služba (service) - je systém, který poskytuje jeden nebo více tzv. endpointů. WCF služba je vnímaná okolním světem jako kolekce endpointů. WCF službu tvoří tři základní části:

- třída služby, která implementuje poskytovanou službu
- hostované prostředí
- jeden nebo více endpointů

Komunikaci klienta se službou posíláním zpráv znázorňuje následující obrázek



Obrázek 5.2: Komunikace klienta se službou

Endpoint (koncový bod) – místo pro příjem a odesílání zpráv se skládá ze tří částí

- adresa
- binding
- kontrakt

Pro jednodušší zapamatování se používá následující zápis:

Endpoint = Address + Binding + Contract = ABC

Myslím, že tento zápis nejstručněji a nejvýstižněji charakterizuje filozofii WCF. Endpoint vytvoříme v konfiguračním souboru následovně:

```
<endpoint address="http://158.196.141.100:9000/Courses"
binding="basicHttpBinding" bindingConfiguration="Courses"
contract="eLogikaCourses.CourseFacade"/>
```

- Adresa (address) – specifikuje, kam jsou zprávy zasílány
- Binding (binding) – jedná se o komunikační mechanismus a popisuje, jak mají být zprávy posílány. Mezi základní faktory, které binding specifikuje, patří způsob přenosu (HTTP, TCP apod.) a kódování (text, binary). Binding taky zahrnuje zabezpečení (security session) a podporu transakcí.
- Kontrakt (contract) - specifikuje sadu zpráv, se kterými se může komunikovat
- Hosting – služba musí být hostována v nějakém procese. Host je aplikací, která kontroluje životní cyklus služby.
- Kanál (channel) - je prostředí, v kterém se přenášejí zprávy. Předtím, než si dvě aplikace začnou vyměňovat zprávy, musí se mezi nimi vytvořit kanál směrem k endpointu.
- Metadata - popisují službu, přičemž tento popis používají externí systémy, aby dokázaly komunikovat se službou.

5.5 Komunikace s Raspberry PI

Mikropočítač Raspberry PI nabízí následující funkce:

- Čtení dat z RFID čtečky, která je k mikropočítači připojena
- Tiskový server
- Komunikace s mobilními zařízeními platform iOS, Android, Windows Phone

Komunikace s tiskovým serverem je založena na protokolu TCP/IP, kdy postup k vytvoření komunikace vypadá následovně:

- Klient se připojí do sítě
- Klient vytvoří logiku pro odesílání a přijímání síťové komunikace skrze TCP/IP
- Klient se spojí se serverem na daném portu v dané síti
- Klient komunikuje se serverem formou streamu
- Klient odešle testovací zprávu ve formátu JSON

Komunikace se serverem při tisku testů

- Zašleme na server REQUEST uvítací zprávu v následující podobě: {method_type:"WELCOME_NOTE" message:"Hi!" };
- Následně čekáme na RESPONSE zprávu o úspěšném navázání spojení: {method_type:"WELCOME_NOTE" message:"Succesfully connected"}
- Následně můžeme zasílat REQUEST zprávy s požadovanými parametry: {method_type:"PRINT_PDF_FILE" isQR:0 PDF_data:"BASE64EncodedData"}
Pokud je hodnota isQR=0 jedná se o tisk testu, v opačném případě se jedná o tisk QR kódu
- Po vykonání požadovaných operací je nutno ukončení spojení a to provedeme následovně: {method_type:"END_COMMUNICATION"}

Komunikace se serverem při snímání studentských karet RFID

Navázání a ukončení spojení je totožné s předchozím případem. Skenování probíhá takto:

- Zašleme na server REQUEST zprávu, kterou zahájíme skenování karet: {method_type:"BEGIN_SCAN"}
- Po úspěšném sejmutí karty nám přijde RESPONSE zpráva s informacemi o majiteli karty {method_type:"BEGIN_SCAN" scanned_id:"ID_UIZVATELE"}
- Skenování ukončíme zasláním REQUEST zprávy: {method_type:"END_SCAN"}

Chybová hlášení

V jakékoliv fázi komunikace může přijít zpráva o chybě:

error_message:"Custom error message"

5.6 Zvolené technologie

Společnost Microsoft poskytuje několik verzí svých vývojářských balíčků. Já jsem vsadil logicky na tu nejaktuálnější verzi, tj. SDK 8.0, která je s kombinací C#/XAML pro mě tou nejlepší možnou volbou.

XAML

XAML (Extensible Application Markup Language) je značkovací jazyk vyvinutý společností Microsoft, primárně určený k popisu grafického uživatelského rozhraní. Přestože se možnosti jeho použití během posledních let značným způsobem rozšířily (může být použit například k serializaci objektů, ukládání dokumentů apod.), jeho hlavní oblast využití byla ve WPF aplikacích a nově také ve Windows Phone a Windows 8 aplikacích.

Myšlenkou stojící za vývojem tohoto jazyka byla snaha oddělit vývoj uživatelského rozhraní (který může být prováděn například specializovanými grafiky) od vývoje aplikační logiky. V průběhu času vzniklo několik specializovaných vývojových nástrojů pro editaci XAML souborů (například již zmíněný Expression Blend), určených především pro grafické designéry.

XAML je založen na XML struktuře elementů, které jsou do sebe vzájemně zanořovány. Jednotlivé elementy poté přímo odpovídají .NET objektům v rámci aplikace a programátor tak s nimi může v kódu pracovat (XML elementy jsou mapovány na instance CLR objektů a XML atributy jsou mapovány na jejich vlastnosti).

Aby bylo možné pomocí jazyka XAML navrhovat uživatelská rozhraní dle návrhového vzoru MVVM, je potřeba využít dvou velmi důležitých technologií: Data Binding a Routed Commands.[8]

C#

Jazyk C# je objektově orientovaný programovací jazyk, který vytvořila Microsoft zároveň s platformou NET. První zmínky o něm propukly na veřejnost v roce 2000 jako součást Visual Studio. Je to vlastně zjednodušená, vylepšená a čistě objektová verze programovacího jazyka C++. C# se využívá hlavně k tvorbě databázových programů, webových aplikací, webových služeb, formulářových aplikací ve Windows, Windows Phone apod.[10]

6 Aplikace a její implementace

Při vývoji aplikací pro platformu Windows Phone 8 se často využívá tzv. asynchronní programování. Smyslem asynchronního programování je zabránit zpomalení, či dokonce přerušení odezvy na povely od uživatele bez ohledu na to, zda jsou zadávané dotykovými gesty, pomocí myši či klávesnice. Jinými slovy, aplikace se vůči uživateli nesmí jevit jako nečinná.

Například aplikace, která stahuje informace z Internetu (ze systému eLogika), může strávit několik sekund čekáním na požadované informace. Při použití synchronní metody pro toto vlákno je aplikace blokována, dokud tato metoda nenačte údaje. Aplikace nebude reagovat na interakci s uživatelem, což může způsobit nespokojenost uživatele, případně až pocit, že aplikace "zamrzla". Mnohem lepší je použít právě asynchronní přístup, kde aplikace, zatímco čeká na dokončení asynchronní operace, běží a reaguje na změny v uživatelské rozhraní.[8]

Typickými příklady pro asynchronní volání jsou:

- Odesílání a přijímání údajů do/z internetu
- Zobrazení oznámení
- Práce se soubory

6.1 Přihlašování

Tato stránka aplikace používá klasického portrétového rozložení grafického uživatelského rozhraní a obsahuje dva přihlašovací cykly. V prvním z nich po úspěšném zadání přihlašovacích údajů a stisknutí tlačítka **Ověřit**, aplikace kontroluje uživatele za pomoci webové služby Users, kde metoda DBAuthenticate hledá uživatele v databázi systému eLogika. Pokud je uživatel nalezen a přihlašovací údaje odpovídají těm z databáze, jsou uživateli zobrazeny univerzity, role a akademické roky, které mu přísluší.

Zde byly využity webové služby Years, Shools a Roles a metody na vyhledání pomocí identifikačního čísla uživatele. Po zvolení univerzity, role a akademického roku se může uživatel přihlásit. Následně mu bude zobrazeno příslušné uživatelské rozhraní.

XAML:

```
<Grid x:Name="ContentPanel" Margin="74,36,76,34"
Grid.RowSpan="2">

<Grid.RowDefinitions><RowDefinition Height="430*" />
<RowDefinition Height="270*" /></Grid.RowDefinitions>

<TextBox Name="TextboxLogin" Margin="0,165,0,0"
BorderBrush="#FF54317E" SelectionForeground="#FFB4B4B4" />

<Button Name="ButtonAuthorization" Content="Ověřit"
Margin="110,326,0,0" Width="220" ... Click="Button_Click" />
```

```
<PasswordBox Name="PasswordBox" HorizontalAlignment="Left"
Margin="0,254,0,0" ... BorderBrush="#FF54317E" Height="72"/>

<TextBlock HorizontalAlignment="Left" Margin="10,138,0,0"
Text="Přihlašovací jméno:" ... Height="27" Width="200"/>

<TextBlock HorizontalAlignment="Left" Margin="10,232,0,0"
Text="Heslo: .... Height="27" Width="92"/>

<toolkit:ListPicker x:Name="listPickerUniversity"
ExpansionMode="FullScreenOnly" SelectionMode="Single"
FullModeHeader="Výběr univerzity">

    </Grid>
</Grid>
```

Ukázka tzv. kódu v pozadí při ověřování uživatele v systému eLogika:

Dojde k načtení údajů o uživateli z ovládacích prvků aplikace a následnému předání těchto údajů příslušné metodě instance webové služby User. Poté v tzv. handleru, který byl definován, bude možno přistupovat k výsledkům dané metody. V tomto případě se výsledek uloží připravené instance třídy User.

Volání:

```
var login = TextboxLogin.Text;
var pass = PasswordBox.Password;

// Check user start
UserFacadeClient user = new UserFacadeClient();
user.DBAuthenticateCompleted += user_DBAuthenticateCompleted;
user.DBAuthenticateAsync(login, pass);
```

Handler:

```
User user = e.Result;
UserIdentifiaciton.userID = user.IdUzivatel;
UserIdentifiaciton.userName = user.Jmeno + " " + user.Prijmeni;

// Load school start
SchoolFacadeClient school = new SchoolFacadeClient();
school.GetSchoolsByIdUzivatelCompleted +=
school_GetSchoolsByIdUzivatelCompleted;
school.GetSchoolsByIdUzivatelAsync(UserIdentifiaciton.userID);
```


The image displays two versions of the 'eLogika' login interface. The left version is the default state with empty input fields. The right version shows the form after data entry. The labels and their corresponding values are as follows:

Label	Value
Přihlašovací jméno:	ger116
Heslo:	[Masked]
Univerzita:	Vysoká škola Báňská - Tec
Uživatelská role:	Garant
Akademický rok:	2013/2014

Obrázek 6.1: Přihlašovací obrazovka

6.2 Rozhraní role student

Využívá panoramatického rozložení uživatelského rozhraní a nabízí uživateli možnost autentizovat se na aktivitě a vytvoření či aktualizaci stávající fotografie. Může k tomu využít buď přední fotoaparát, nebo vybrat fotografii z galerie telefonu. K autentizaci se používají informace o uživateli získané v procesu přihlašování. K aktualizaci fotografie je zapotřebí služby Users a metody UpdateUserPhoto, která přijme fotografii jako pole bajtů.

Převod na pole bajtů:

```
public static byte[] ConvertToBytes(BitmapImage bitmapImage){
    byte[] data = null;
    using (MemoryStream stream = new MemoryStream())
    {
        WriteableBitmap wb = new WriteableBitmap(bitmapImage);
        wb.SaveJpeg(stream, wb.PixelWidth,
wb.PixelHeight, 0, 100);
        stream.Seek(0, SeekOrigin.Begin);
        data = stream.GetBuffer();
    }
    return data;
}
```



Obrázek 6.2: Rozhraní role student

6.2.1 Rozhraní fotoaparátu

Jelikož je předpokládáno pořizování snímku na výšku, je zvolen opět klasický portrétový model uživatelského rozhraní, které využívá kamery umístěné na čelní straně mobilního telefonu. Uživateli bude po dotyku na displej pořízena fotografie, kterou si může prohlédnout na své domovské obrazovce.

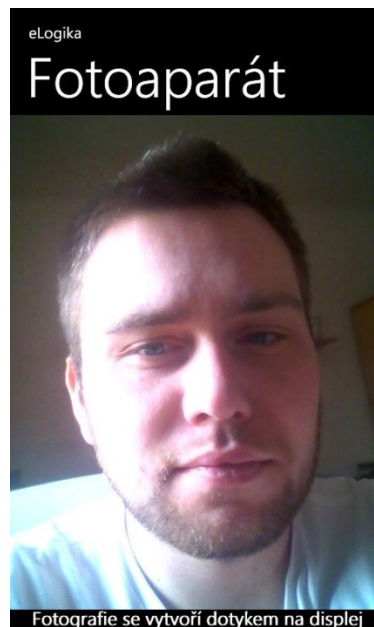
XAML:

```
<Grid x:Name="LayoutRoot" Background="Transparent">
<Canvas x:Name="viewfinderCanvas" Height="640" Width="480"
Tap="viewfinder_Tapped" Margin="0,138,0,46" Grid.RowSpan="2">
    <Canvas.Background><VideoBrush x:Name="viewfinderBrush">
        <VideoBrush.RelativeTransform>
            <CompositeTransform
                x:Name="viewfinderTransform"
                CenterX="0.5" CenterY="0.5"
                Rotation="-90"/>
        </VideoBrush.RelativeTransform>
    </VideoBrush></Canvas.Background>
</Canvas>
```

Kód v pozadí - zachycení a následný přístup k snímku:

```
private void viewfinder_Tapped(object sender,
System.Windows.Input.GestureEventArgs e) {
    if (cam != null) {
        try {
            cam.CaptureImage();
        }
        catch (Exception ex) {
            this.Dispatcher.BeginInvoke(delegate() {
                txtMessage.Text = ex.Message;
            });
        }
    }
}

private void cam_CaptureImageAvailable(object sender,
Microsoft.Devices.ContentReadyEventArgs e) {
    Deployment.Current.Dispatcher.BeginInvoke(delegate() {
        txtMessage.Text = "Ukládám fotku";
    });
    DecodeImage(e.ImageStream, 270);
}
```



Obrázek 6.3: Rozhraní fotoaparátu

6.3 Rozhraní role tutor/garant

Opět je využito panoramatického rozložení grafického uživatelského rozhraní. Vyučující zde musí v první fázi vybrat termín aktivity, na které chce studenty autentizovat a zvolit, co chce při ověřování tisknout. Poté má na výběr dvě možnosti autentizace.

Možnost NFC očekává, příjem NFC zprávy v podobě přihlašovacího jména studenta, dle kterého si zjistí, zdali je student přihlášen na zvoleném termínu (služba Dates, metoda IsInTermin) a zobrazí vyučujícímu fotografii, kterou je nucen přijmout nebo zamítnout. Když přijme, je zahájen tisk dle zvoleného nastavení. Tisk se provádí na základě komunikace s tiskovým serverem, jehož komunikaci jsem nastínil v předchozí kapitole mé práce.

Pod autentizace pomocí NFC spadá ještě jedna funkce. Jedná se o schválení fotografie vyučujícím.

K druhé funkci ICIC, jak název napovídá, je zapotřebí studentské karty s RFID. Kdy po úspěšné komunikace karty s čtečkou (připojena k mikropočítači Raspberry PI) aplikace obdrží přihlašovací jméno studenta, který opět ověří se zvoleným termínem, a poté je zahájen tisk.



Obrázek 6.4: Rozhraní role tutor/garant

Kód v pozadí - příjem NFC zprávy od druhého zařízení

```
_subscriptionId = _proximitydevice.SubscribeForMessage
("Windows.NokiaExample", (device, message) =>{
    Deployment.Current.Dispatcher.BeginInvoke(() =>{
        studentID = Convert.ToInt32(message.DataAsString);
    })
})
```

StudentID je následně předáno instanci třídy RFIDFilter, která slouží k získání testu a QR kódu studenta.

Kód v pozadí - získání testu a QR kódu

1. K úspěšnému získání testu, je nutno vytvořit objekt `RFIDFilter`, který se používá jako je parametr k získání odpovědi ze serveru systému eLogika.

```
RFIDFilter rfid = new RFIDFilter{  
    CourseId = (from c in courses where c.Nazev ==  
        vyberKurzu.SelectedItem.ToString() select c.IDKurzInfo).First(),  
    DateId = (from d in terms where d.Nazev ==  
        vyberTerminu.SelectedItem.ToString() select t.IDTermin).First(),  
    SchoolInfoId = UserIdentifiaciton.userSchoolInfoID,  
    StudentId = studentID,  
    TestId = (from t in tests where t.Nazev ==  
        vyberTestu.SelectedItem.ToString() select t.IDTest).First()  
};
```

2. Zisk QR kódu si žádá o dva parametry (`SchoolInfoID`, získané `StudentID`). Dále je třeba zavolat příslušné metody webové služby `RFIDService`.

```
RFIDServiceClient rfid = new RFIDServiceClient();  
// Get QR code start  
rfid.GetStudentQRCodeCompleted +=  
    rfid_GetStudentQRCodeCompleted;  
rfid.GetStudentQRCodeAsync(studentId, schoolInfoId);  
// Get test start  
rfid.GetStudentTestCompleted += rfid_GetStudentTestCompleted;  
rfid.GetStudentTestAsync(rfidFilter);
```

3. Výsledek obdržíme přes handler v obojím případě v poli bajtů, které musíme převést na formát, který tiskový server umí zpracovat (`BASE64EncodedData`).

```
private void rfid_GetStudentTestCompleted(object sender,  
    GetStudentTestCompletedEventArgs e)  
{  
    byte[] testB = e.Result;  
    if (testInBytes != null){  
        test = Convert.ToBase64String(testB, 0, testInBytes.Length);  
    }  
}
```

4. Získaný test dále posíláme na tiskový server, se kterým navážeme spojení následovně.

```
// Connect

await this.StreamSocket.ConnectAsync(new HostName(HOST_NAME),
PORT);

// Start communication

await Write(GetMessage(1));

resultMessage = await Read();

// Result

JsonData resultMessage;

return resultMessage.method_type;
```

5. Nyní čekáme na odpověď z tiskového serveru. Pokud obdržíme požadovaný tvar odpovědi, můžeme zahájit odesílání tiskových požadavků, které jsou popsány v předchozí kapitole.

Závěr

Hlavním cílem mé bakalářské práce bylo vyvinout aplikaci pro operační systém Windows Phone 8, kterou bude moci uživatel využívat při autentizaci na testech a aktivitách systému eLogika, což požadují za splnění v celém svém rozsahu.

Největší porci času jsem strávil nad seznámením se se systémem eLogika a pochopením jeho fungování. Při následném vývoji aplikace jsem se nesetkal s ničím, co by mi činilo větší potíže. Opět šlo hlavně o to, zorientovat se v množství metod webových služeb, kterých systém poskytuje obrovské množství a vybrat tu správnou.

Tato práce byla pro mě obrovským přínosem, co se týče praktického ověření mých teoretických znalostí. Poprvé jsem si například vyzkoušel práci s WCF službami, využil asynchronního programování a naučil se pracovat v týmovém prostředí Microsoft Visual Studio (Team Foundation Server).

Co se týče možného rozšíření mé aplikace, tak bych navrhl napojení na službu pro rozpoznávání obličejů, kdy by fotka, kterou si chce uživatel (student) nahrát na svůj účet byla porovnáвана s jeho fotografií ze systému Edison. Byla by však nutná provázanost s tímto systémem, což momentálně není.

Další potencionální využití by mohly přinést NFC tagy, které by byly k dispozici na všech testech a aktivitách systému eLogika. V jejich paměti by byl nastaven odkaz na obchod s aplikacemi příslušné platformy s vyfiltrovaným seznamem aplikací využívající systém eLogika. Tento způsob by určitě prospěl k lepší propagaci mobilních aplikací, následnému zvýšení počtu stažení a k navýšení počtu zpětných vazeb a postřehů, které jsou velmi důležité pro další vývoj těchto aplikací.

Seznam obrázků

<i>Obrázek 2.1: Funkce role student.....</i>	<i>- 3 -</i>
<i>Obrázek 2.2: Funkce role tutor</i>	<i>- 3 -</i>
<i>Obrázek 2.3: Funkce role garant.....</i>	<i>- 4 -</i>
<i>Obrázek 4.1: Vývoj uživatelské rozhraní systému Windows Phone</i>	<i>- 12 -</i>
<i>Obrázek 4.2: Struktura projektu</i>	<i>- 14 -</i>
<i>Obrázek 5.1: Aktivitní diagram autentizace</i>	<i>- 19 -</i>
<i>Obrázek 5.2: Komunikace klienta se službou</i>	<i>- 20 -</i>
<i>Obrázek 6.1: Přihlašovací obrazovka.....</i>	<i>- 25 -</i>
<i>Obrázek 6.2: Rozhraní role student.....</i>	<i>- 26 -</i>
<i>Obrázek 6.3: Rozhraní fotoaparátu</i>	<i>- 27 -</i>
<i>Obrázek 6.4: Rozhraní role tutor/garant</i>	<i>- 28 -</i>

Seznam tabulek

Tabulka 1: Srovnání bezdrátových komunikačních technologií.....	- 8 -
---	-------

Seznam příloh

Příloha A:	Sekvenční diagram přihlašování	I
Příloha B:	Sekvenční diagram autentizace.....	II
Příloha C:	Třídní diagram aplikace	iii

Součástí BP je CD.

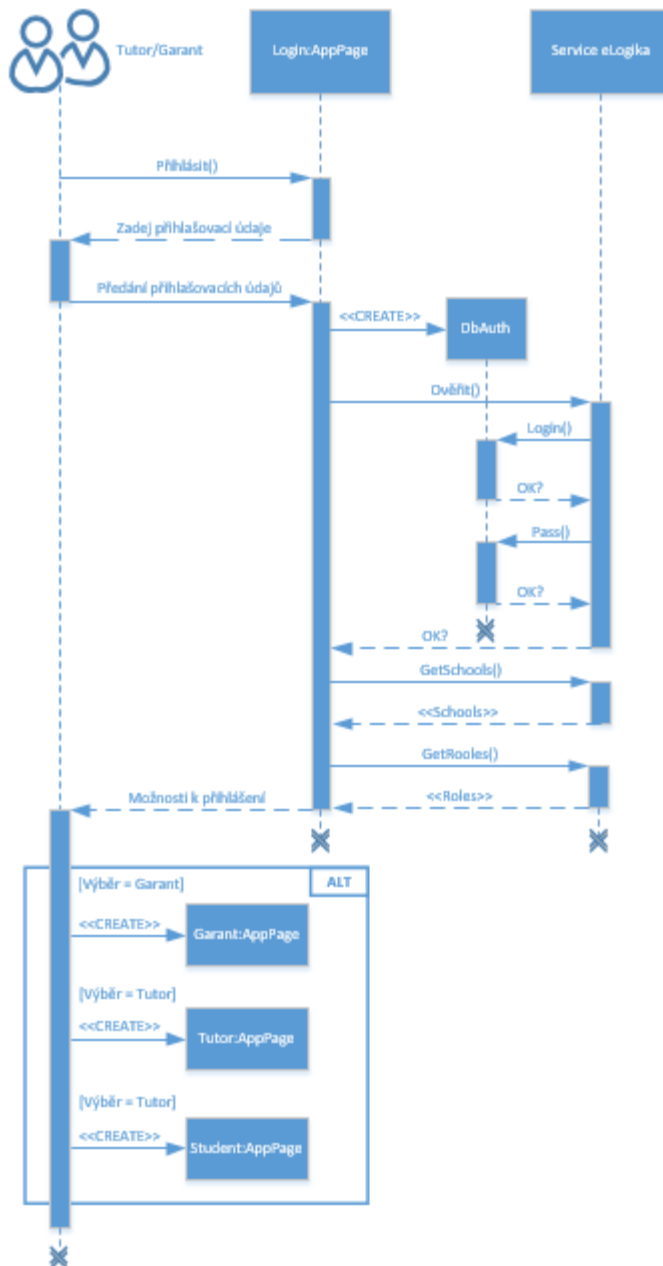
/sources	Zde jsou uloženy zdrojové aplikace
/diagrams	Zde jsou k nalezení vývojové diagramy
/screens	Zde jsou k nalezení obrázky uživatelského rozhraní aplikace
/thesis	V tomto adresáři je přiložen tento text

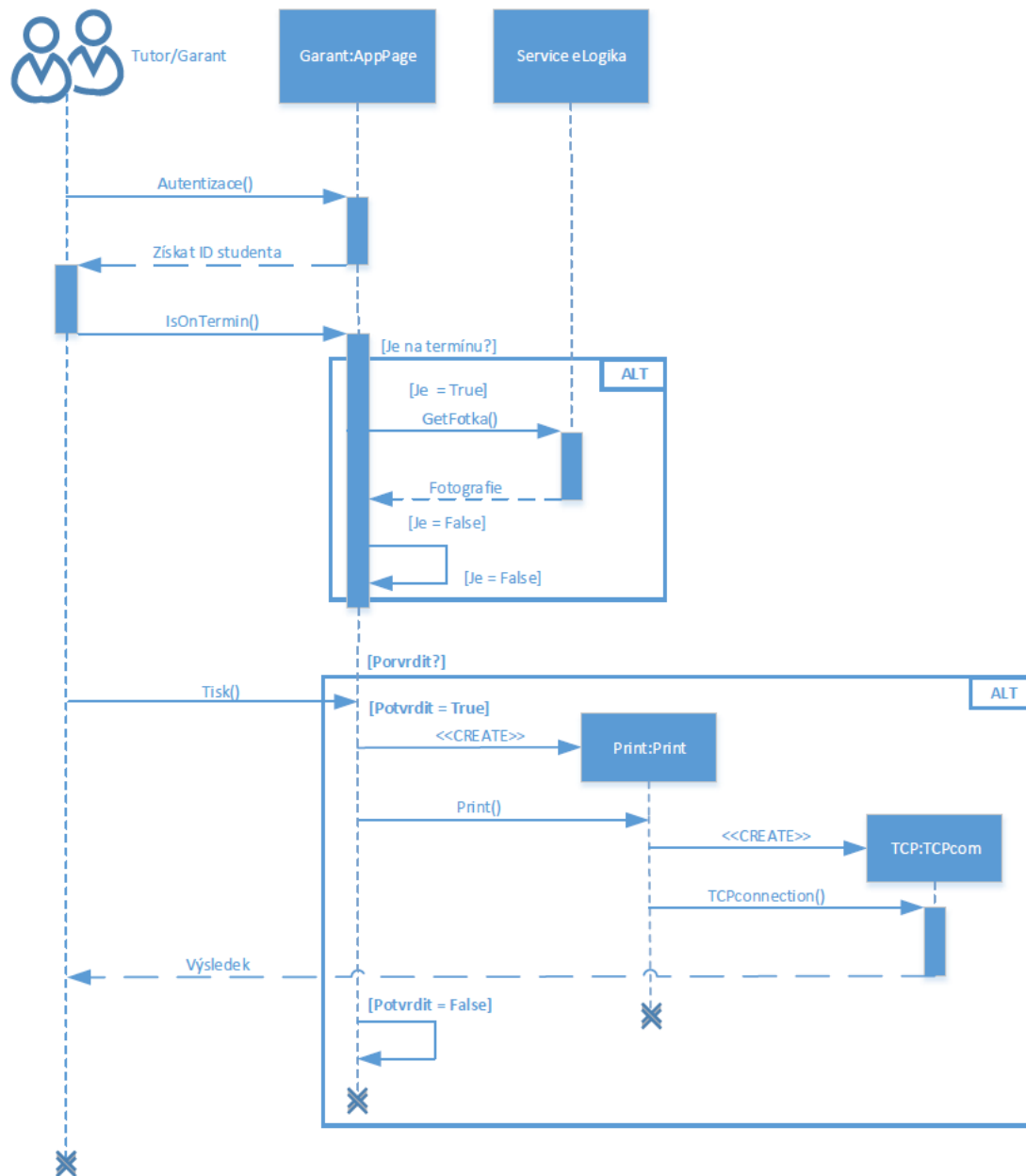
Použitá literatura

- [1] HERNAS Vojtěch. Systém eLogika a e-learningová podpora výuky, diplomová práce, VŠB-TU Ostrava 2011
- [2] NFC Forum [online]. [cit. 2014-05-01]. Dostupné z: <http://nfc-forum.org/what-is-nfc/about-the-technology/>
- [3] COSKUN Vedat, OK Kerem, OZDENIZCI Busra. Near Field Communication(NFC): From Theory to Practice. ISBN 978-1-119-97109-2
- [4] IGOE Tom, COLEMAN Don, JEPSON Brian. Beginning NFC. ISBN 978-1-4493-7206-4
- [5] BOBEK, Juraj. Identifikace v systému eLogika za pomoci RFID, diplomová práce, VŠB-TU Ostrava 2012
- [6] GLOVER Bill, BHATT Himanshu. RFID Essentials. ISBN 978-0596009441
- [7] SZOSTAK Tomasz. Windows Phone 8 Application Development Essentials. ISBN 978-1-84969-677-7
- [8] LACKO Luboslav. Vývoj aplikací pro Windows 8.1 a Windows Phone. ISBN 978-80-251-3822-9
- [9] KLEIN Scott. Profesional WCF Programming. ISBN 978-1-4571-0702-3
- [10] GLYNN Jay, Watson Karli, SKINNER Morgan, ROBINSON Simon, NAGEL Christian, ALLEN K.Scot, CORNES Ollie, GREENVOSS Zach, HARVEY Burton. C# Programujeme profesionálně. ISBN 978-80-251-2401-7
- [11] KABELOVÁ Alena, DOSTÁLEK Libor. Velký průvodce protokoly TCP/IP a systémem DNS. ISBN 978-80-251-2236-5

Sekvenční diagram přihlašování

Příloha A: *Sekvenční diagram přihlašování*





Příloha C: *Třídní diagram aplikace*

